

Network delivery of interactive entertainment synchronized to playback of audio recordings

Patent number: JP2001521642T

Publication date: 2001-11-06

Inventor:

Applicant:

Classification:





- International: G10K15/02; G06F13/00; G06F17/30; G11B27/00; G11B27/10; H04H1/00; H04N7/173; H04L29/08

- european:

Application number: JP19980544286T 19980415

Priority number(s): US19970838082 19970415; WO1998US07660 19980415

Also published in:

 WO98/00001
 WO98/00001
 EP1000000
 EP1000000
 US5900000

more

Abstract not available for JP2001521642T

Abstract of correspondent: **US5987525**

Entertainment content complementary to a musical recording is delivered to a user's computer by means of a computer network link. The user employs a browser to access the computer network. A plug-in for the browser to control an audio CD or other device for playing the musical recording. A script stored on the remote computer accessed over the network is downloaded. The script synchronizes the delivery of the complementary entertainment content with the play of the musical recording.

(19) 日本国特許庁 (J P)

(12) 公表特許公報 (A)

(11) 特許出願公表番号
特表2001-521642
(P2001-521642A)

(43) 公表日 平成13年11月6日 (2001. 11. 6)

(51) Int.Cl. ⁷	識別記号	F I	テ-マ-ド* (参考)
G 1 0 K 15/02		G 1 0 K 15/02	
G 0 6 F 13/00	6 5 0	G 0 6 F 13/00	6 5 0 B
	1 1 0		1 1 0 F
	4 1 9		4 1 9 B
G 1 1 B 27/00		G 1 1 B 27/00	A

審査請求 未請求 予備審査請求 有 (全 40 頁) 最終頁に続く

(21) 出願番号 特願平10-544286
 (86) (22) 出願日 平成10年4月15日 (1998. 4. 15)
 (85) 翻訳文提出日 平成11年10月15日 (1999. 10. 15)
 (86) 国際出願番号 P C T / U S 9 8 / 0 7 6 6 0
 (87) 国際公開番号 W O 9 8 / 4 7 0 8 0
 (87) 国際公開日 平成10年10月22日 (1998. 10. 22)
 (31) 優先権主張番号 0 8 / 8 3 8 , 0 8 2
 (32) 優先日 平成9年4月15日 (1997. 4. 15)
 (33) 優先権主張国 米国 (US)
 (81) 指定国 EP (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), AU, CA, JP

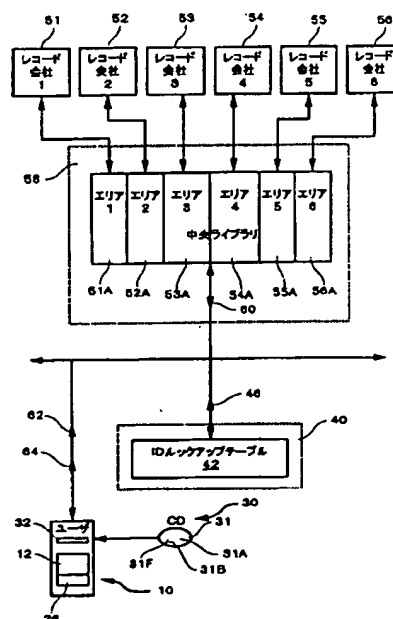
(71) 出願人 イオン・インコーポレーテッド
 アメリカ合衆国 カリフォルニア州94960,
 サン アンセルモ, サンフランシスコ・ブルヴァード216
 (72) 発明者 ロバーツ・デイル・タイソン
 アメリカ合衆国 カリフォルニア州94965,
 サウサリート, スペンサー・アヴェニュー164
 (72) 発明者 グリーンバーグ・アン・イー
 アメリカ合衆国 カリフォルニア州94960,
 サン アンセルモ, サンフランシスコ・ブルヴァード216
 (74) 代理人 弁理士 土井 健二 (外1名)

(54) 【発明の名称】 音声記録物を補足する対話型エンタテインメントのネットワーク配信

(57) 【要約】

音楽記録物を補足するエンタテインメントコンテンツが、コンピュータネットワークリンクによって、ユーザのコンピュータヘデリバリ (配信) される。ユーザは、そのコンピュータネットワークにアクセスするために、ブラウザを用いる。そのブラウザに対するプラグインは、音楽記録物を演奏するための音声CDやその他のデバイスをコントロールすることができる。ネットワーク越しにアクセスされた、遠隔にあるコンピュータ上に格納されたスクリプトが、ダウンロードされる。そのスクリプトが、補足エンタテインメントのデリバリ (配信) を、音楽記録物の演奏に同期させる。

図 1



前記遠隔の装置は、第二のユーザ装置であることを特徴とする方法。

9. 第一のあらかじめ記録された媒体からのコンテンツを遠隔の装置からネットワークを介して配信される補足コンテンツに同期させる装置において、

ビデオ表示手段を有するローカルな電子装置と、

前記ローカルな電子装置に接続され、又は一体化した、あらかじめ記録されたコンテンツのプレイヤーと、

遠隔の装置からネットワークを介して前記ローカル装置に、補足コンテンツ及び同期情報を転送する手段と、

前記第一のあらかじめ記録された媒体のコンテンツの明示を前記補足コンテンツに同期させるように、前記同期情報を使って、前記あらかじめ記録されたコンテンツのプレイヤーを制御する手段を備えることを特徴とする装置。

10. 請求の範囲9において、さらに、

前記第一のあらかじめ記録された媒体の状態情報を、前記第一のあらかじめ記録された媒体からネットワークを介して前記遠隔の装置に転送する手段と、

あらかじめ記録された媒体の状態情報と前記同期情報を使って、前記補足コンテンツの明示を制御する手段とを備えることを特徴とする装置。

11. 請求の範囲9において、

前記第一のあらかじめ記録された媒体は、CDであることを特徴とする装置。

12. 請求の範囲9において、

前記第一のあらかじめ記録された媒体は、DVDであることを特徴とする装置。

13. 請求の範囲9において、

前記第一のあらかじめ記録された媒体は、音声コンテンツを含むことを特徴とする装置。

14. 請求の範囲9において、

前記第一のあらかじめ記録された媒体は、音声及びビデオコンテンツを含むことを特徴とする装置。

15. 請求の範囲9において、

前記第一のあらかじめ記録された媒体は、音楽を含むことを特徴とする装置。

16. 請求の範囲9において、
前記第一のあらかじめ記録された媒体を分析する手段と、
前記第一のあらかじめ記録された媒体の前記コンテンツに基づいて識別コード
を計算する手段と、
前記識別コードに基づいて、前記遠隔の装置から補足コンテンツ及

び同期情報を転送する手段とを備えることを特徴とする装置。

17. 請求の範囲9において、
前記遠隔の装置は、ネットワークサーバであることを特徴とする装置。
18. 請求の範囲9において、
前記遠隔の装置は、第二のユーザ装置であることを特徴とする装置。

【発明の詳細な説明】

音声記録物を補足する対話型エンタテインメントのネットワーク配信

背景

発明の分野

本発明は、コンピュータネットワークの分野に属し、特に、CD ROM、音声記録物、その他配布媒体に関連したサービスをユーザへ提供するためのネットワークプロトコルの使用に関する。

関連技術

ここ数年来、オンラインサービスが爆発的な成長を遂げ、エンタテインメントの一つの主要な形態となってきた。この新しいエンタテインメントと並んで、音楽記録物などのより伝統的な形態も大規模に消費され続けている。

従来、音楽記録物は、一つの部屋に集まった小人数のグループによって聞かれる場合が多い。その場合、音楽は部屋を音響的に満たすが、その音楽に関連した視覚的コンテンツは殆どなく、その記録物に対する介入行為も、本質的にはトラックの選択と、音量の設定やエコライザの適用といった簡単な音の調整に限られている。このような伝統的な聞き方は、約一世紀前の78回転レコードの時代と変わっていない。

伝統的な音楽記録物の製作方法は、前記伝統的な聞き方を補足している。記録物は、数多くの録音段階で製作され、注意深いミキシングと編集を受けた後に、一般の人々へリリースされる。この点で、記録物は、今日では音声CDは、作者、演奏者、プロデューサー、及びレコーディングエンジニアによってデザインされた最終的な音をできるだけ忠実に記録することを目的とした固定的な形態であるといえる。

音楽ビデオは、記録物のトラックに視覚的コンテンツを結合するこ

とにより、伝統的な音楽記録物の聞き方を補ってきた。しかしながら、実際問題として、音楽ビデオはユーザコントロールが欠如しているという問題をもってテレビ放映されてきており、消費者の対話性や参加という点では貢献してきていない。

オンラインサービスは、既に記録された素材と携わる経験を豊かにする機会を提供する。本発明は、この望みを実行することのできるコンピュータプログラム、システム、及びプロトコルを提供するものである。

発明の要旨

そこで、本発明の一つの目的は、製作者がインターネットなどのオンラインサービスを用いて、配布媒体記録物を補足するエンタテインメントをデリバリ（配信）することのできるコンピュータプログラム、システム、及びプロトコルを提供することである。

また、本発明の別の目的は、前記補足エンタテインメントを消費者に対して有意義に対話型とすること、即ち消費者もその聞き方のクリエイタになることのできるコンピュータプログラム、システム、及びプロトコルを提供することである。

更に、本発明の別の目的は、前述の目的が、オンラインサービスに対する標準の継続的な進化に対応できる柔軟性を残したまま、既存の、特にインターネット上の環境とプログラムとのインテグレーションを達成するようにデザインされた方法によって達成されることである。

本発明の一つの側面は、CDなどの配布媒体の製造者と販売者に対し、顧客との関係を維持し、かつ強化するための手段を提供することである。レコード会社は、その会社が市場で持っているCDに対する補足コンテンツの中央ライブラリへのダウンロードと周期的な更新を行う。本発明のソフトウェアは、ユーザのWebブラウザへのプラグインとして機能し、レコード会社のCDと共に、ユーザをユーザのCDに対して適切な中央ライブラリの特定のセクションへ導く。

本発明の別の側面は、レコード会社のCDに関連したユーザの聞く習慣を代表する、いわゆる“CDウォッチャー”データが、補足コンテンツがネットワークコネクションを介してユーザへデリバリされた時に、レコード会社へ渡されることである。

その代わりに、中央ライブラリに貢献し維持しているレコード会社は、補足コンテンツのために中央ライブラリへアクセスしたすべてのユーザの聞く習慣へアクセスすることができる。

本発明の更に別の側面は、遠隔にあるホストコンピュータ上で動くコンピュータプログラムに、ユーザのコンピュータ上のCDプレーヤ、DVDプレーヤなどの配布媒体プレーヤをコントロールさせることを許すソフトウェアが提供されることである。（便宜上、全ての配布媒体を“CD”と、また、DVDプレーヤやその類似品などの全ての配布媒体プレーヤを“CDプレーヤ”と呼ぶ）このソフトウェアは、遠隔にあるホストコンピュータに、CDプレーヤに関するアクションを起こすことと、CDプレーヤのフロントパネルのボタンや他のCDプレーヤコントロールプログラムなど他の手段によってユーザが起こしたアクションを検知すること、の両方を許すようにデザインされている。発明のこの側面は、その記録物が現代一般に広く使われている形態であるCDに固定的に記録される時のCDコンテンツに対する補足エンタテイメントの仕組みの一部分である。

本発明の別の側面は、対話型コンテンツを含んだ視覚的コンテンツを、一つのオンラインサービスを介して、音楽記録物のコンテンツのデリバリと同期してデリバリできることである。例えば、そのような視覚的コンテンツを、ユーザのコンピュータの音声CDやその他の配布媒体の演奏と同期させてもよい。その視覚的コンテンツは、例えば音楽ビデオのように、音楽記録物とテーマがリンクしている。

本発明の更に別の側面は、数多くのトラックを含むCDや他の配布媒体に対して実質的にユニークな識別子を決定する若しくは割り当てる方法が提供されることである。ユニークな識別子は、補足コンテン

ツをデリバリするソフトウェアに、補足コンテンツに対応する実際に正しいCDを確認させることができ、CDや他の配布メディアの演奏と共に補足コンテンツをデリバリするのに有益である。例えば、補足コンテンツがRosary Sonatas of Heinrich Ignaz Franz Biberを伴うようにデザインされているときに、ユーザのプレーヤのCDや他の配布媒体がMary Poppinsのサウンドトラックであった場合、多分機能を果たさないだろう。また、このユニークな識別子は、CDや他の配布媒体がプレミアムWebエリアへアクセスするキーとして用いられることを可能とする。更に、このユニークな識別子は、ユーザのマシンにあるCDや他の配布媒体に対応

するWebのエリアへユーザを導くことができる。

本発明の更に別の側面は、一般に“チャットルーム”として知られている非常に人気のあるオンラインサービスを、チャットルームにいる全ての人が聞いているCD記録物へのリンクにより、強化（拡充）することである。遠隔にあるホストコンピュータは、複数の遠隔地に位置する配布媒体プレーヤをコントロールすることができる。チャットルームでの経験は、今日オンラインサービスで存在しているように、お互いが相手を識別しうる環境における伝統的なface-to-faceの社会的な出会いと比較し、肉体と遊離した特性を持つ。今日、チャットユーザの唯一共通の経験は、コンピュータスクリーン上を飛び回るチャットの言葉と、ユーザアイコン(“avatars”)、あるいはスクリーン上の小さな空間を占める他の視覚的コンテンツである。チャットルームと共に音楽記録物を使用することは、ある程度伝統的な社会的出会いの共有された環境での経験へ戻す可能性を打ち開く。更に、音楽記録物のような共有コンテンツは、チャットシーカーがある特定のタイプの記録物に対する共有の興味によってグループ化することを可能にする注目点を提供する。

図面の簡単な説明

図1は、本発明の最初の具現物のネットワーク図である。

図2は、図1に示される具現物のフロー図である。

図3は、本発明が動作する環境のブロック図である。

図4は、本発明の一つの側面に従った補足コンテンツのブロック図である。

図5は、本発明の同期コードのフローチャートである。

図6は、本発明をチャットルームへ連結するための操作の手順のフローチャートである。

好ましい実施の形態の説明

図1及び図2を参照すると、本発明の実施の形態は、CDを再生するユーザを、データ交換用の遠隔なホストと接続する。ブロック11において、ユーザは、遠隔なホストにリンクし、クライアント26のダウンロードを要求する。ダウンロードを受信するために、ユーザは、いくつかの基本的な情報、名前、eメール

、チャットネームなどを用意しなければならない。ブロック 11 Aにおいて、その登録情報は、クライアント 26と交換される。ブロック 10 Aにおいて、ユーザのコンピュータ 10は、クライアント 26を有するブラウザのようなコンピュータプログラム 12を実行する。ブロック 30 Aにおいて、プレイヤ 32にCD 30を挿入すると、クライアント 26による動作が起きる。クライアント 26は、プレイヤ 32を制御して、CD 30を走査する。CD 31は、時間コード 31 A及び他の暗号化されたデータ 31 Bを含んでもよい。ブロック 34で、クライアント 26は、CDの走査の結果を使って、実質的にユニークなCDのID 31を計算する。いくつかのCDは、IDのテキストファイル又はIRCコードを含んでもよく、共通の規格はなく、かくして、ID計算技術は、いかなるCDに対しても有効なIDをもたらす。この技術は、CD 30の実質的にユニークなパターンマッチ(match) 31 Fを生成するために、CD 30の内容の一部をサンプリングし、そのサンプルを使うことに

よって、パターンマッチ(match) 31 Fを生成してもよい。クライアント 26がCDのID 31を計算している間、ブロック 36において、利用人口データ 62が、クライアント 26によって収集され、一時的に格納される。

利用人口データ 62は、最も利用されたトラック、利用のトータル時間、最も利用されたCD、コンピュータ 10が稼働時間の平均長さ、ロードされたソフトウェア、最も利用されたソフトウェア、クライアントと同時に実行されるソフトウェアなどを含むCD利用プロファイルを含む。利用人口データ 62は、このCD利用プロファイルに限定されない。

CDのID 31の計算は、クライアント 26を刺激して、コンピュータプログラム 12にCDのID 31のためのローカルキャッシュをチェックさせる。CDのID 31がローカルキャッシュにない場合、ブロック 38において、クライアント 26は、ルックアップサーバ 40にリンクする。コンピュータプログラム 12がまだ実行されていない場合、クライアント 26は、それを始めてもよい。ルックアップサーバ 40へのリンクが確立すると、クライアント 26は、CDのID 31をルックアップサーバ 40に送信する。ルックアップサーバ 40は、CDのID 31をテーブル

42と比較する。テーブル42は、CDのIDを、関連するコンテンツ及び補足コンテンツを有する特定のアドレスにリンクさせるルックアップテーブルである。他の情報、例えば、タイミング及び制御データ、電気製品の優待券、広告、及びタイミング及び制御データを有するビデオのようなボーナスコンテンツがテーブル42に含まれてもよい。ブロック44において、ルックアップサーバ40は、CDのID31の受信にตอบสนองして、アドレス46のような情報をユーザに送信する。ユーザに送信された情報は、ルックアップサーバ40に送信されたユーザ人口データ及び利用人口データ62に基づいても、基づかなくともよい。ブロック48において、クライアント26は、アドレス46へのリンクを確立する。アドレス4

6は、サイト51Aのようなプレミアム(premium)サイト又は予約サイトであってもよく、この場合、CDのID31は、パスワードとして作用してもよい。

本発明の一つの側面において、それぞれレコード会社1-6のようなコンテンツ供給者51-56は、ウェブ上に中央ライブラリを保持する。また、コンテンツ供給者は、広告主、CD小売り者、及び他のコンテンツ権利保有者を含んでもよい。中央ライブラリ50は、ルックアップサーバ40のような単一のサーバ上にあってもよく、また、分散されていてもよい。中央ライブラリ50は、アドレス46のようなテーブル42内のアドレスによってリンクするサイト51Aのような補足コンテンツサイトを含む。各コンテンツ供給者51-56は、新しいCDがリリースされると、かれらのサイトのコンテンツを変更してもよく、追加のサイトへの新しいリンクを追加してもよい。新しいサイトが追加されると、新しいCDのID及びリンクされるアドレスがテーブル42に追加される。

ブロック49Aにおいて、サイト51Aを主催するサーバ58は、ユーザのコンピュータ10に補足コンテンツを転送する。ユーザがサイト51Aにリンクすると、クライアント26は、格納された利用人口データ62及びCDステータスデータ64をルックアップサーバ40又はサーバ58に送信する。ブロック66において、クライアント26は更新を続行し、クライアント26が実行されている限り、利用人口データ62及びCDステータスデータ64をルックアップ又はサーバ58に送信する。閉ループの更新70によって、コンテンツ供給者51-56の

ようなコンテンツ供給者は、リアルタイム又はほとんどリアルタイムに、どのCDが使用中か、CD要素の利用の相対的な頻度、及び関連する情報を見ることができる。また、閉ループの更新70は、ユーザのコンピュータ10上で再生する配布されたメディアの制御を提供する。ブロック68におけるCDの削除及び変更は、クライアントをブロック30Aに戻す。

別の実施の形態において、本発明は、ワールドワイドウェブ上で機能する。ウェブ上のHTTPプロトコルは、一般的な接続用プロトコルの上で実行される。一般的な接続用プロトコルは、今日では、一般的に、Douglas E. Comerの"internetworking with TCP/IP (3d ed. 1995)"に記述されるTCP/IPである。しがしなから、ここに述べられる発明は、特別な種類のネットワークソフトウェア又はハードウェア上で実行されるHTTPに限定されない。発明の原理は、HTTPと競合する、又はHTTPに取って代わる、遠隔の情報にアクセスするための他のプロトコルに適応する。

図3を参照すると、ユーザは、コンピュータ10の場所に座り、ブラウザ又は他のクライアントソフトウェアのようなコンピュータプログラム12を実行する。ブラウザは、サーバ16のようなサーバとして参照される他のコンピュータにHTTP要求14を送信する。要求において、資源として参照されてサーバ上で有効であるデータの特定の項目は、ユニフォームリソースロケータ(URL)によって参照され、符号(character)は、以前にBurners-Leeらによって定義された特定の形式で配列する(string)。URLは、サーバの識別符号及びサーバ内のデータの特定の項目の識別符号の両方を含む。要求に反応して、サーバは、ユーザのブラウザに応答18を返し、ブラウザは、それらの応答に反応し、一般的に、ユーザにある種類のコンテンツを表示する。

応答のコンテンツ部分は、20及び22ページのようなハイパーテキストマークアップ言語(HTML)で表現される「ウェブページ」であってもよい。その言語は、ビットマップ形式の画像及び(アンカー及びハイパーリンクとして知られる)リンクが散在するテキストを構成するコンテンツを表現するための言語である。リンクは、ユーザの意志によってブラウザが更なる要求を送信するURLである。

また、応答は、ブラウザによって翻訳されるより複雑なコマンドを含んでもよく、例えば、コマンドは、後述の図4で議論されるように、動画を生じさせるためのコマンドである。HTMLそれ自身は複雑な

コマンドを定義せず、むしろ、複雑なコマンドは、別に定義されたスクリプト言語に属すると考えられ、その中で現在一般的な2つの言語は、ジャバスクリプト (JavaScript) 及びVBスクリプト (VBScript) である。

スクリプト言語で書かれたコードを使ったブラウザの機能の拡張に加えて、変換された (compiled) コードによるブラウザの機能の拡張も可能である。そのような変換されたコードは、プラグインと呼ばれる。プラグインを書くための正確なプロトコルは、特定のブラウザに依存する。マイクロソフトのブラウザのためのプラグインは、アクティブX (ActiveX) 制御の名前で呼ばれる。

アクティブXは非常に複雑かもしれない。本発明に関連して有利に利用されるプラグインは、マクロメディア (Macromedia) のショックウェーブ (shockwave) である。それは、サーバの応答の部分である動画 (animations) をダウンロードし、再生することを可能にする。ショックウェーブは、リンゴ (Lingo) と呼ばれる自己のスクリプト言語を定義する。リンゴスクリプトは、ショックウェーブのプラグインが再生できるダウンロード可能な動画内に含まれる。ショックウェーブの動画の一般的な形式は、連続したフレームを構成するタイムラインであり、多数の視覚的な物体がタイムライン内の特定のフレームで現れ、動き、消える。ショックウェーブでより複雑な効果を達成するために、リンゴスクリプトは、あらかじめ定義された視覚的なオブジェクトに加えてインボーク (invoke) されてもよい。

本発明の現在好ましい実施の形態は、コマンドプラグイン24と呼ばれるプラグインを利用し、コマンドプラグイン24は、スクリプト言語に、命令する能力及び記録されたものであるCD記録物の再生を詳しく監視する能力を提供する。コマンドプラグインは、少なくとも、以下の基本的な機能を提供し、監視する。

- (1) 再生の開始及び停止
- (2) 現在のトラックとトラック内の位置の取得

- (3) トラック及びトラック内の位置のシーク
- (4) 音量の取得及び設定
- (5) CDに関する情報の取得（例えば、トラック数、トラックの長さ、トラック間の区切り (pause)）
- (6) CDドライブの性能に関する情報の取得

他の機能が提供され、監視されてもよく、他の機能は、基礎のオペレーティングシステムのサービスが適応可能であることによつてのみ限定される。監視される機能は、サーバ40及び又はサーバ58のようなサーバに転送される利用人口データに含まれる。

コマンドプラグインは、C++のような従来のプログラミング言語で書かれてもよい。プラグインは、マイクロソフトのアクティブXオブジェクトに必要とされる規格のようなプラグインのための存在する規格に従わなければならない。情報を取得し、コマンドプラグインがスクリプト言語で有効にされる機能を実行するために、コマンドプラグインは、制御及び再生する音楽が記録されたものに関する情報を提供する機能に依存する。記録されたものの正確なソース (source) に依存する。本発明の一つの形態のように、もし、再生される記録物がコンピュータのCDプレイヤー内の音声CDである場合、そして、ブラウザがマイクロソフトのウィンドウズ3.1又はウィンドウズ95又はウィンドウズCEの下で実行される場合、これらの機能は、Win3.2アプリケーションプログラミングインターフェースの一部を形成するMCI機能である。これらの機能は、例えば、マイクロソフトのWin3.2のプログラムの参考書に載っている。異なる機能が、音声受信装置のストリーミング (streaming) によつて提供されてもよく、例えば、受信装置は、MPEGのような適当な音声暗号化フォーマットで接続するネットワークを介してユーザのコンピュータ内に入ってくる音声を捕まえる。

コマンドプラグインの実行について注意すべき重要なポイントは、プラグインが実行する、例えばシークのような動作は、秒オーダーの

時間がかかる可能性があることである。コマンドプラグイン24にとつて、その間隔の間、コンピュータ10の制御を保留することは好ましくないので、冗長な

動作が行われるときはいつでも、コマンドプラグイン24がコンピュータ10の制御をブラウザに渡し、共通のスクリプト言語を使用した非同期イベント取り扱い能力を介した動作の結果を報告することが重要である。

上記のコマンドプラグインが提供する機能の要約、プラグインの書き方（例えば、アクティブXオブジェクトの書き方）の一般的な知識、及びCDの再生を制御するための関連するアプリケーションプログラミングインターフェース（例えば、Win32のMCI）の知識が与えられることで、当業者は、容易に且つ過大な経験なしに、実際に働くコマンドプラグインを開発することができる。この理由によって、いかにコマンドプラグインが実行されるかについての更なる詳細は、ここに提供されない。

上記にリストされた機能をスクリプト言語に提供するコマンドプラグインの存在は、CDのコンテンツを補足するエンタテインメントが作られる基礎である。特に、スクリプト言語によって、補足コンテンツの表示をCDで発生するイベントと同期させる方法を、この基礎の上に案出することができる。

図4及び5を参照すると、CDへの補足コンテンツの同期は、以下のように進行する。例えば、補足コンテンツ60は、フレームF1-Fn及びスクリプト72を含むショックウェーブの動画のような動画によって提供される。補足コンテンツ60は、サーバ58からダウンロードされ、ショックウェーブのプラグインによって、ユーザに表示される。このダウンロードは、動画が表示される前に行われてもよいし、又は、代わりに、ネットワークへのユーザの接続が、適当な速度のダウンロードを維持するのに十分速ければ、動画が表示されながら行われてもよい。ダウンロードは、ショックウェーブのプラグインそれ自身により提供される機能である。

ショックウェーブの動画が再生されながら、リンゴ(Lingo)スクリプトのようなスクリプト72は、一つのフレームFnが表示を終了する毎に実行する。リンゴスクリプトは、動画のフレームとCDコンテンツのセグメントとの間に存在すべき関係の記述を含み、その関係は、トラック番号及び時間によって識別される。リンゴスクリプトは、上記のコマンドプラグインによって、CDの再生がどのトラッ

ク及び時間であるかを決定する。それから、CDのその部分に対応する動画のフレームを決定するために、記述を参照する。現在のフレームが決定されたフレームの一つでない場合、リンゴスクリプトは、動画のタイムラインをリセットし、動画は、CDの現在位置に対応するフレームで再生を開始する。これは、例えば、ネットワークからのダウンロードが遅れたため、ユーザのコンピュータが最大限の速度で動画を再生するための周期を欠いたため、また、ユーザがCDを早送りしてしまったために、動画がCDに遅れる場合、視覚的コンテンツが追いつくことを可能にする。

図4を参照すると、同期のアルゴリズムは、個々のフレーム又は隣接するフレームのグループを制御してもよい。補足コンテンツ60は、フレームF1-Fn及びスクリプト72を含む。ブロック200で、各フレームFn又はフレームのグループとCD30の特定のセグメントとの間の対応関係が設定される。ブロック205での動画の各フレームFnの終わりに、ブロック210で、CD30の位置が決定される。ブロック215で、CD30の位置が、次の連続するフレームが属するフレームのグループに対応する記録物のセグメント内であるかどうかを決定するテストが実施される。CD30の位置がそのセグメント内である場合、ブロック230で、動画の再生は次のフレームに進む。CD30の位置がそのセグメント内でない場合、ブロック220及び225で、アニメーションの再生は、CDの位置に対応するフレームに進む。

本発明の更なる側面は、周遊 (touring) モードである。周遊モー

ドのクライアント26において、サーバ58のような遠隔の装置は、プレイヤ26の制御データが散在する補足コンテンツ60のデリバリを介して、一人又はそれ以上のユーザの装置を制御し、一人又はそれ以上のユーザに、CD30のような特定の配布された媒体の案内される周遊を提供する。

本発明の更なる側面は、コマンドプラグイン24を使って、ユーザのCDプレイヤ32内にあるCDのユニークな識別子 (CDのID31)を設定するための技術を提供する能力である。ユニークな識別子は、(例えば、1秒の1/75単位で測定される)トラックの数及び長さに基づいてもよく、その場合、識別子は、これらの長

さの連鎖となる。

しかしながら、実際は、多少短い識別子を有することが望ましい。そこで、ユニークな識別子は、好ましくは、1秒の1/4単位のような、かなり粗い単位で表現されるトラック長さの連鎖である。

プログラムAは、2つの音声CDが正確に又はほぼ同じであるかどうかを決定するのに適したファジーな比較アルゴリズムのためのC言語で書かれたソースコードである。ファジーな比較アルゴリズムは、以下のように進行する。比較される2つの音声CDの各々に対して、数ミリ秒内に記録物内の全トラックの長さを決定する。それから、全てのトラックの長さを8ビットずつ右に移動する(shift)。その結果、28=256による打ち切り分割(truncating division)を実行する。それから、一致合計及び一致エラーの2つの数を積算しながら、両記録物を1トラックずつ通過する。これらの数は、両方ともに、比較の開始時にゼロに初期化される。各トラックについて、比較される第一のCDのトラックの移動した長さにより一致合計を加算し、2つのCDのトラックの移動した長さの間の差の絶対値により一致エラーを加算する。より小さいトラック数でCDの最後のトラックに到達すると、他のCDのトラックに継続し、トラックの移動した長さによる一致合計及び一致エラー両方を加算する。トラックを通過するこれらのステップの後、アルゴリズムは、一致数により一致エラーを除算し、1からその商を

減算し、その差を、どれくらい2つのCDが一致しているかの指標である割合に変換する。

プログラムBは、2つの音声CDが正確に同じであるかどうかを決定するのに適した比較アルゴリズムのためのC言語で書かれたソースコードである。アルゴリズムは、トラック数、トラック長さ及びトラックの開始及び終了時間から、8バイト値を生成する。上位4バイトは、数ミリ秒以内で表わされる、全てのトラックの開始及び終了時間を合計することによって得られる。下位4バイトは、数ミリ秒以内で表わされる全てのトラックの長さを合計し、その合計を左に10ビット移動し、トラックの数を追加することによって得られる。

CDのID3 1は、データベースの鍵として利用されてもよい。サイト5 2 Aのよ

うなサイトは、CDについての情報のデータベースを維持してもよく、例えば、レコード会社2によって発行されるすべてのCDについての情報は、レコード会社のサイト上で保持される。ユーザのために、この情報にたどり着く(navIgate)ための種々の代替的な方法がある。例えば、ユーザは、コンテンツのテーブルとしての多くのハイパーリンクを含むウェブページを使用することができ、また、ユーザは、従来の検索エンジンを使用することもできる。本発明のCDのID31により可能になる検索の第三の方法は、ユーザが情報を探しているCD(例えば、CD30)をプレイヤ32に置くことをユーザに促すウェブページであることである。ドライブの中にCDの存在が検知されると、ウェブページ内のスクリプトはCD30に対応するCDのID31を計算し、それをサーバ58に送信する。それから、サーバ58は、CDのID31に基づいてデータベースから引き出されたCDについての情報を表示する。この情報は、CDに関するウェブアドレス(URL)(例えばアーティストのホームページのウェブアドレス)、CD上の歌の名前のような簡単なデータ、そして、(例えばバンドの)写真、作品、動画及びビデオクリップを含む補足的なエンターテインメントを含んでもよい。ユーザがコンピュータにCDを挿入す

るとき、(i)ウェブブラウザがまだ実行されていない場合に開始し、(ii)ブラウザがCDのユニークな識別子を計算し、そのユニークな識別子からURLを抽出し、(iii)ブラウザがURL上でHTTP取得処理を行うように、順序を配列することもできる。

音楽記録物のユニークな識別子の別の適用は、ウェブのプレミアムエリアへ入るための鍵としCDを利用することである。現在、人々が申し込みにより許可されるウェブの特別(premium)な場所がある。ユニークな識別子に基づいた許可の簡単な形式は、ウェブの特定の場所にアクセスする前に、特定のCD、又は特定の会社によって発行されたCD、又は特定のバンド又はアーティストの音楽を含むCDをユーザのCDドライブに置くことを要求することである。これは、コマンドプラグインによって提供される機能をインボーク(involve)し、ユニークな識別子を計算するスクリプトによって簡単に達成される。

本発明の別の側面は、チャットルームと音楽記録物との接続である。目的は、

チャットルームの全ての参加者に、ほぼ同じ時間に同じ音楽を提供することである。

チャットサービスのための従来のネットワークプロトコルは、Internet Relay Chat(IRC)のJ. Oikarinen & D. Reedによって記述されたInternet Relay Chat Protocol(Internet Request For Comments No. 1459, 1993)である。このプロトコルにおいて、ある人がチャットサーバのクライアントになると、その人はチャットルームに名前を送信する。チャットサーバは、その全てのクライアントからのメッセージを受信し、あるクライアントから送られたメッセージを、クライアントとして同じルームに接続される他のクライアントに中継する。クライアントが送信するメッセージは、典型的に、クライアントを実行しているユーザによってタイプされ、クライアントが受信するメッセージは、典型的に、クライアントを実行しているユーザに読むために表示される。

本発明の好ましい実施形態では、チャットクライアントは、チャッ

トプラグインと呼ばれるであろうプラグインによってカスタマイズ(customized)される。チャットクライアントは、以下のように、ブラウザによって開始される(図5参照)。ユーザは、ブラウザによって、中央ウェブページ(ボックス300)に接続する。中央ウェブページは、ダウンロードされるとき、ユーザがプレイヤーにCDを挿入することを要求する(ボックス305)。CDのユニークな識別子が計算され、中央ウェブページのスクリプトのコマンドに従って、上記の制御プラグインを使って、サーバに返信される(ボックス310)。それから、サーバは、ユニークな識別子を使って、それがCD専用(focused on)のチャットルームを持っているかどうかを決定する。このステップは、既知の技術を使って、ユニークな識別子をデータベース内に探すことによって実行されてもよい。データベースに接続するウェブページには、例えば、December & Ginsburg, supra, chapter 21のような膨大な文学(literature)が存在する。CD専用(focused on)のチャットルームが存在する場合、又は作られた場合、サーバは、チャットルームの名前に応答して、ブラウザは、チャットルームのクライアントとして、ユーザのコンピュータ上でチャットクライアントを開始する(ボックス320)。

チャットルームの名前は、サーバによって、CDが他のチャットルームのクライアントの装置で再生しているトラック、トラックが再生を開始した時間、さらに、CDが再生している音量についての情報を含むように設定される。チャットクライアントのプラグインはその情報を使用して、ユーザのコンピュータ内のCDを、他のチャットルームのクライアントの装置で再生しているCDにほぼ同期して再生するように設定することをコントロールプラグインに指示する(ボックス 3 2 0)。

チャットルーム内の各ユーザは、各ユーザの装置で再生するCDを制御することができる。制御動作により、チャットプラグインは、メッセージを、行われた制御動作を記述するチャットサーバに送信する

(ボックス 3 2 5)。例えば、そのようなメッセージは、CDの位置の変更、音量の変更、別のCDに取り替えるためのCDの取り出しを示してもよい。他のユーザの装置で進行するチャットプラグインは、この種のメッセージを見ると、上述の制御プラグインを使って、他のユーザの装置で(できるだけ)その動作を繰り返す(reuplicate)。

本発明の更なる側面では、特定の音楽記録物専用(focused on)のチャットルームは、特定のトラックを選択するための投票手続きを認めてもよい。簡単は投票手続きは、各チャットプラグインにおいて、それが2つの同一の連続した変化メッセージを見つけると、前の段落で述べられた種類のメッセージの変化に従って動作することである。これは、再生されているトラックを変えるために、二人のユーザがトラックを変える必要があることを意味する。2という数字は、それより大きい数字に置き換えられてもよい。

本発明の更なる側面において、チャットのユーザに配信されるメッセージは、手でのタイプよりテキストファイルから作られてもよい。これは、あらかじめ記録された経験(experience)を、チャットユーザのグループのために再生することを可能にする。そのような技術は、音声CDのあらかじめ記録され、語り付きの周遊(touring)を作るのに使用されてもよい。

上記された実施形態の重要な利点は、Internet Relay Chat又は同様の最小限

のチャットサービスを提供するプロトコルによって要求される最小限の機能をサポートするいかなるチャットサーバソフトウェアに使用できることである。要求される追加のソフトウェアは、CD情報のデータベースに接続したチャットクライアントプラグイン及び中央ウェブページにある。

プログラムA

```
/*
 * FUZZY CD ID
 * (c) 1996-1998 ION Inc.
 *
 * by Ty Roberts
 */

#include <stdio.h>
#include <stdlib.h>
#include <time.h>

struct fuzzyCDid {
    short          numberTracks; // start time in
    milliseconds
    unsigned short fuzzlength[100];
};

typedef struct fuzzyCDid fuzzyCDid, *fuzzyCDidPtr;

// structure of a cd track with all times stored in milliseconds

struct cdtrack {
    long beginMs; // start time in milliseconds
    long endMs;   // end time in milliseconds
    long lengthMs; // length in milliseconds
};

typedef struct cdtrack cdtrack, *cdTrackPtr;

struct cd {
    short    numberTracks;
    cdtrack  track[100];
};

typedef struct cd cd, *cdPtr;

void CreateFuzzyId( fuzzyCDidPtr fid, cdPtr cd );

float    FuzzyMatch( fuzzyCDidPtr fid1, fuzzyCDidPtr fid2 );

// SUBROUTINES

void CreateFuzzyId( fuzzyCDidPtr fid, cdPtr cd )
{
```

```

long    i;

// first copy in the number of tracks
fid->numberTracks = cd->numberTracks;

for(i=0;i<fid->numberTracks;i++) {
    // shift left and create a MSB length thats not exact
    fid->fuzzlength[i] = (short)(cd->track[i].lengthMs>>8);
}

float    FuzzyMatch( fuzzyCDidPtr fid1, fuzzyCDidPtr fid2 )
{
    long    fidmatcherr = 0, fidmatchtotal = 0;
    short    i, trackcnt;
    float    matchpercent;

    // find the larger number of tracks
    trackcnt = fid1->numberTracks<fid2->numberTracks ? fid2->numberTracks : fid1->numberTracks;

    // cycle thru the tracks accumulating error and total
    comparedtimes
    for(i=0;i<trackcnt;i++) {
        if ((i < fid1->numberTracks) && (i < fid2->numberTracks)) {
            fidmatcherr += abs(fid1->fuzzlength[i] - fid2->fuzzlength[i]);
            fidmatchtotal += fid1->fuzzlength[i];
        } else if (i >= fid2->numberTracks) {
            fidmatcherr += fid1->fuzzlength[i];
            fidmatchtotal += fid1->fuzzlength[i];
        } else if (i >= fid1->numberTracks) {
            fidmatcherr += fid2->fuzzlength[i];
            fidmatchtotal += fid2->fuzzlength[i];
        }
    }

    if (fidmatcherr > 0) {
        matchpercent = 100 -
        (((float)fidmatcherr/(float)fidmatchtotal) *100);
    } else {
        matchpercent = 100;
    }
    return matchpercent;
}

```

```

void main(void)
{
    short i;
    float    matchpercent;

    // create global structures for two complete cds with up to
100 tracks
    cd    cd2id;
    fuzzyCDid fidcd2id;
    cd    cdFromDB;

    fuzzyCDid fidcdFromDB;

    printf ("Test #1 will compare two CDs that are exactly the
same\n\n");

    // put in some test values for the cd track lengths
    // since these are in ms, its basically 60000 = 1 minute

    cd2id.track[0].lengthMs = 121323;
    cd2id.track[1].lengthMs = 234565;
    cd2id.track[2].lengthMs = 566437;
    cd2id.track[3].lengthMs = 245120;
    cd2id.track[4].lengthMs = 20000;
    cd2id.track[5].lengthMs = 120386;
    cd2id.track[6].lengthMs = 323453;
    cd2id.numberTracks = 7;

    for(i=1;i<cd2id.numberTracks;i++) {
        printf ("CD #1: Track = %d  length in minutes = %f\n",
            i, (float)cd2id.track[i].lengthMs/60000.0);
    }
    printf("\n");

    cdFromDB.track[0].lengthMs = 121323;
    cdFromDB.track[1].lengthMs = 234565;
    cdFromDB.track[2].lengthMs = 566437;
    cdFromDB.track[3].lengthMs = 245120;
    cdFromDB.track[4].lengthMs = 20000;
    cdFromDB.track[5].lengthMs = 120386;
    cdFromDB.track[6].lengthMs = 323453;
    cdFromDB.numberTracks = 7;

    for(i=1;i<cdFromDB.numberTracks;i++) {
        printf ("CD #2: Track = %d  length in minutes = %f\n",

```

```

        i, (float)cdFromDB.track[i].lengthMs/60000.0 );
    }

    CreateFuzzyId( &fidcd2id, &cd2id );
    CreateFuzzyId( &fidcdFromDB, &cdFromDB );

    matchpercent = FuzzyMatch( &fidcd2id, &fidcdFromDB );
    printf ("The cd's matchpercent was computed as=%f",
matchpercent);
    printf ("\n");
    printf ("\n");

    printf ("Test #2 will compare two cd that are nearly the
same\nexcept they have diffent # of tracks \n");

    // put in some test values for the cd track lengths
    // since these are in ms, its basically 60000 = 1 minute
    cd2id.track[0].lengthMs = 121323;
    cd2id.track[1].lengthMs = 234565;
    cd2id.track[2].lengthMs = 566437;
    cd2id.track[3].lengthMs = 245120;
    cd2id.track[4].lengthMs = 20000;
    cd2id.track[5].lengthMs = 120386;

    cd2id.track[6].lengthMs = 323453;
    cd2id.numberTracks = 7;

    for(i=1;i<cd2id.numberTracks;i++) {
        printf ("CD #1: Track = %d length in minutes = %f\n",
            i, (float)cd2id.track[i].lengthMs/60000.0 );
    }
    printf ("\n");
    cdFromDB.track[0].lengthMs = 121323;
    cdFromDB.track[1].lengthMs = 234565;
    cdFromDB.track[2].lengthMs = 566437;
    cdFromDB.track[3].lengthMs = 245120;
    cdFromDB.track[4].lengthMs = 20000;
    cdFromDB.track[5].lengthMs = 120386;
    cdFromDB.numberTracks = 6;

    for(i=1;i<cdFromDB.numberTracks;i++) {
        printf ("CD #2: Track = %d length in minutes = %f\n",
            i, (float)cdFromDB.track[i].lengthMs/60000.0 );
    }

    CreateFuzzyId( &fidcd2id, &cd2id );

```

```

        CreateFuzzyId( &fidcdFromDB, &cdFromDB );
        matchpercent = FuzzyMatch( &fidcd2id, &fidcdFromDB );

        printf ("The cd's matchpercent was computed
as=%f",matchpercent);
        printf ("\n");
        printf ("\n");
        printf ("Test #3 will compare two cd that are not the
same\n\n");

        // put in some test values for the cd track lengths
        // since these are in ms, its basically 60000 = 1 minute
        cd2id.track[0].lengthMs = 34213;
        cd2id.track[1].lengthMs = 334565;
        cd2id.track[2].lengthMs = 231423;
        cd2id.track[3].lengthMs = 134122;
        cd2id.track[4].lengthMs = 2342;
        cd2id.track[5].lengthMs = 3487;
        cd2id.track[6].lengthMs = 9976;
        cd2id.numberTracks = 7;

        for(i=1;i<cd2id.numberTracks;i++) {
            printf ("CD #1: Track = %d length in minutes = %f\n",
                i, (float)cd2id.track[i].lengthMs/60000.0 );
        }
        printf ("\n");
        cdFromDB.track[0].lengthMs = 121323;
        cdFromDB.track[1].lengthMs = 234565;
        cdFromDB.track[2].lengthMs = 566437;
        cdFromDB.track[3].lengthMs = 245120;
        cdFromDB.track[4].lengthMs = 20000;
        cdFromDB.track[5].lengthMs = 120386;
        cdFromDB.track[6].lengthMs = 323453;
        cdFromDB.numberTracks = 6;

        for(i=1;i<cdFromDB.numberTracks;i++) {
            printf ("CD #2: Track = %d length in minutes = %f\n",
                i, (float)cdFromDB.track[i].lengthMs/60000.0 );
        }

        CreateFuzzyId( &fidcd2id, &cd2id);
        CreateFuzzyId( &fidcdFromDB, &cdFromDB);

        matchpercent = FuzzyMatch( &fidcd2id, &fidcdFromDB );

        printf ("The cd's matchpercent was computed
as=%f",matchpercent);
    }

```

プログラムB


```

/*
 * EXACT MATCH CD ID
 * © 1996-1998 ION Inc.
 *
 * by Ty Roberts
 */

#include <stdio.h>
#include <stdlib.h>
#include <time.h>

struct cdid{
    long id[2];
};

typedef struct cdid cdid, *cdidPtr;

// structure of a cd track with all times stored in milliseconds

struct cdtrack{
    long beginMs; // start time in milliseconds
    long endMs;   // end time in milliseconds
    long lengthMs; //length in Milliseconds
};

typedef struct cdtrack cdtrack, *cdTrackPtr;

struct cd {
    short    numberTracks;
    cdtrack  track[100];
};

typedef struct cd cd, *cdPtr;

void CreateUniqueId( cdidPtr cid, cdPtr cd );

// SUBROUTINES
void CreateUniqueId( cdidPtr cid, cdPtr cd )
{
    long      i, t, n;

    t = 0;
    n = 0;

```

```

        for(i=0;i<cd->numberTracks;i++) {
            // shift left and create a MSB length thats not exact
            t += cd->track[i].lengthMs;
            n += cd->track[i].beginMs + cd->track[i].endMs;
        }
        cid->id[0] = t<<10+cd->numberTracks;
        cid->id[1] = n;
    }

void main(void)
{
    short i;
    short    matchtest;

    // create global structures for two complete cds with up to
100 tracks
    cd    cd2id;
    cdid  cd2UID;

    cd    cdFromDB;
    cdid  cdFromDBUID;

    printf ("Test #1 will compare two cd that are exactly the
same\n\n");

    // put in some test values for the cd track lengths
    // since thes are in ms, its basically 60000 = 1 minute
    cd2id.track[0].beginMs = 0;
    cd2id.track[1].beginMs = 100001;
    cd2id.track[2].beginMs = 231001;
    cd2id.track[3].beginMs = 345001;
    cd2id.track[4].beginMs = 435001;
    cd2id.track[5].beginMs = 460001;
    cd2id.track[6].beginMs = 590001;

    cd2id.track[0].endMs = 100000;
    cd2id.track[1].endMs = 231000;
    cd2id.track[2].endMs = 345000;
    cd2id.track[3].endMs = 435000;
    cd2id.track[4].endMs = 460000;
    cd2id.track[5].endMs = 590000;
    cd2id.track[6].endMs = 690000;
    cd2id.track[0].lengthMs = cd2id.track[0].endMs -
cd2id.track[0].beginMs;

```

```

        cd2id.track[1].lengthMs = cd2id.track[1].endMs -
cd2id.track[1].beginMs;
        cd2id.track[2].lengthMs = cd2id.track[2].endMs -
cd2id.track[2].beginMs;
        cd2id.track[3].lengthMs = cd2id.track[3].endMs -
cd2id.track[3].beginMs;
        cd2id.track[4].lengthMs = cd2id.track[4].endMs -
cd2id.track[4].beginMs;
        cd2id.track[5].lengthMs = cd2id.track[5].endMs -
cd2id.track[5].beginMs;
        cd2id.track[6].lengthMs = cd2id.track[6].endMs -
cd2id.track[6].beginMs;
        cd2id.numberTracks = 7;

        for(i=1;i<cd2id.numberTracks;i++) {
            printf ("CD #1: Track = %d    length inminutes = %f\n",
1, (float)cd2id.track[i].lengthMs/60000.0 );
        }
        printf ("\n");
        cdFromDB.track[0].beginMs = 0;
        cdFromDB.track[1].beginMs = 100001;
        cdFromDB.track[2].beginMs = 231001;
        cdFromDB.track[3].beginMs = 345001;
        cdFromDB.track[4].beginMs = 435001;
        cdFromDB.track[5].beginMs = 460001;
        cdFromDB.track[6].beginMs = 590001;
        cdFromDB.track[0].endMs = 100000;
        cdFromDB.track[1].endMs = 231000;
        cdFromDB.track[2].endMs = 345000;
        cdFromDB.track[3].endMs = 435000;
        cdFromDB.track[4].endMs = 460000;
        cdFromDB.track[5].endMs = 590000;
        cdFromDB.track[6].endMs = 690000;
        cdFromDB.track[0].lengthMs = cd2id.track[0].endMs -
cd2id.track[0].beginMs;
        cdFromDB.track[1].lengthMs = cd2id.track[1].endMs -
cd2id.track[1].beginMs;
        cdFromDB.track[2].lengthMs = cd2id.track[2].endMs -
cd2id.track[2].beginMs;
        cdFromDB.track[3].lengthMs = cd2id.track[3].endMs -
cd2id.track[3].beginMs;
        cdFromDB.track[4].lengthMs = cd2id.track[4].endMs -
cd2id.track[4].beginMs;
        cdFromDB.track[5].lengthMs = cd2id.track[5].endMs -
cd2id.track[5].beginMs;

```

```

        cdFromDB.track[6].lengthMs = cd2id.track[6].endMs -
cd2id.track[6].beginMs;
        cdFromDB.numberTracks = 7;

        for(i=1;i<cdFromDB.numberTracks;i++) {
            printf ("CD #2: Track = %d    length inminutes = %f\n",
i, (float)cdFromDB.track[i].lengthMs/60000.0 );
        }

        CreateUniqueId( &cd2UID, &cd2id );

        printf( "Unique ID for CD #1 = %d%d\n", cd2UID.id[0],
cd2UID.id[1] );

        CreateUniqueId( &cdFromDBUID, &cdFromDB );
        printf( "Unique ID for CD #2 = %d%d\n", cdFromDBUID.id[0],
cdFromDBUID.id[1] );

        matchtest = (cd2UID.id[0] == cdFromDBUID.id[0]) &&
(cd2UID.id[1] == cdFromDBUID.id[1]);

        printf ("The cd's match if result is non zero
matchresult=%d",matchtest);

        printf ("\n");

        printf ("\n");
        printf ("Test #2 will compare two cd that are nearly the
same\nexcept they have diffent # of tracks \n");

        // put in some test values for the cd track lengths
        // since thes are in ms, its basically 60000 = 1 minute
        cd2id.track[0].beginMs = 0;
        cd2id.track[1].beginMs = 100001;
        cd2id.track[2].beginMs = 231001;
        cd2id.track[3].beginMs = 345001;
        cd2id.track[4].beginMs = 435001;
        cd2id.track[5].beginMs = 460001;
        cd2id.track[6].beginMs = 590001;
        cd2id.track[0].endMs = 100000;
        cd2id.track[1].endMs = 231000;
        cd2id.track[2].endMs = 345000;
        cd2id.track[3].endMs = 435000;
        cd2id.track[4].endMs = 460000;
        cd2id.track[5].endMs = 590000;

```

```

cd2id.track[6].endMs = 690000;

cd2id.track[0].lengthMs = cd2id.track[0].endMs -
cd2id.track[0].beginMs;
cd2id.track[1].lengthMs = cd2id.track[1].endMs -
cd2id.track[1].beginMs;
cd2id.track[2].lengthMs = cd2id.track[2].endMs -
cd2id.track[2].beginMs;
cd2id.track[3].lengthMs = cd2id.track[3].endMs -
cd2id.track[3].beginMs;
cd2id.track[4].lengthMs = cd2id.track[4].endMs -
cd2id.track[4].beginMs;
cd2id.track[5].lengthMs = cd2id.track[5].endMs -
cd2id.track[5].beginMs;
cd2id.track[6].lengthMs = cd2id.track[6].endMs -
cd2id.track[6].beginMs;
cd2id.numberTracks = 7;

for(i=1;i<cd2id.numberTracks;i++) {
    printf ("CD #1: Track = %d    length inminutes = %f\n",
i, (float)cd2id.track[i].lengthMs/60000.0 );
}

printf ("\n");
cdFromDB.track[0].beginMs = 0;
cdFromDB.track[1].beginMs = 100001;
cdFromDB.track[2].beginMs = 231001;
cdFromDB.track[3].beginMs = 345001;
cdFromDB.track[4].beginMs = 435001;
cdFromDB.track[5].beginMs = 460001;
cdFromDB.track[6].beginMs = 590001;

cdFromDB.track[0].endMs = 100000;
cdFromDB.track[1].endMs = 231000;
cdFromDB.track[2].endMs = 345000;
cdFromDB.track[3].endMs = 435000;
cdFromDB.track[4].endMs = 460000;
cdFromDB.track[5].endMs = 590000;

cdFromDB.track[0].lengthMs = cd2id.track[0].endMs -
cd2id.track[0].beginMs;
cdFromDB.track[1].lengthMs = cd2id.track[1].endMs -
cd2id.track[1].beginMs;
cdFromDB.track[2].lengthMs = cd2id.track[2].endMs -
cd2id.track[2].beginMs;

```

```

        cdFromDB.track[3].lengthMs = cd2id.track[3].endMs -
cd2id.track[3].beginMs;
        cdFromDB.track[4].lengthMs = cd2id.track[4].endMs -
cd2id.track[4].beginMs;
        cdFromDB.track[5].lengthMs = cd2id.track[5].endMs -
cd2id.track[5].beginMs;
        cdFromDB.numberTracks = 6;

        for(i=1;i<cdFromDB.numberTracks;i++) {
            printf ("CD #2: Track = %d   length inminutes = %f\n",
i, (float)cdFromDB.track[i].lengthMs/60000.0 );
        }

        CreateUniqueId( &cd2UID, &cd2id );
        printf( "Unique ID for CD #1 = %d%d\n", cd2UID.id[0],
cd2UID.id[1] );

        CreateUniqueId( &cdFromDBUID, &cdFromDB );
        printf( "Unique ID for CD #2 = %d%d\n", cdFromDBUID.id[0],
cdFromDBUID.id[1] );

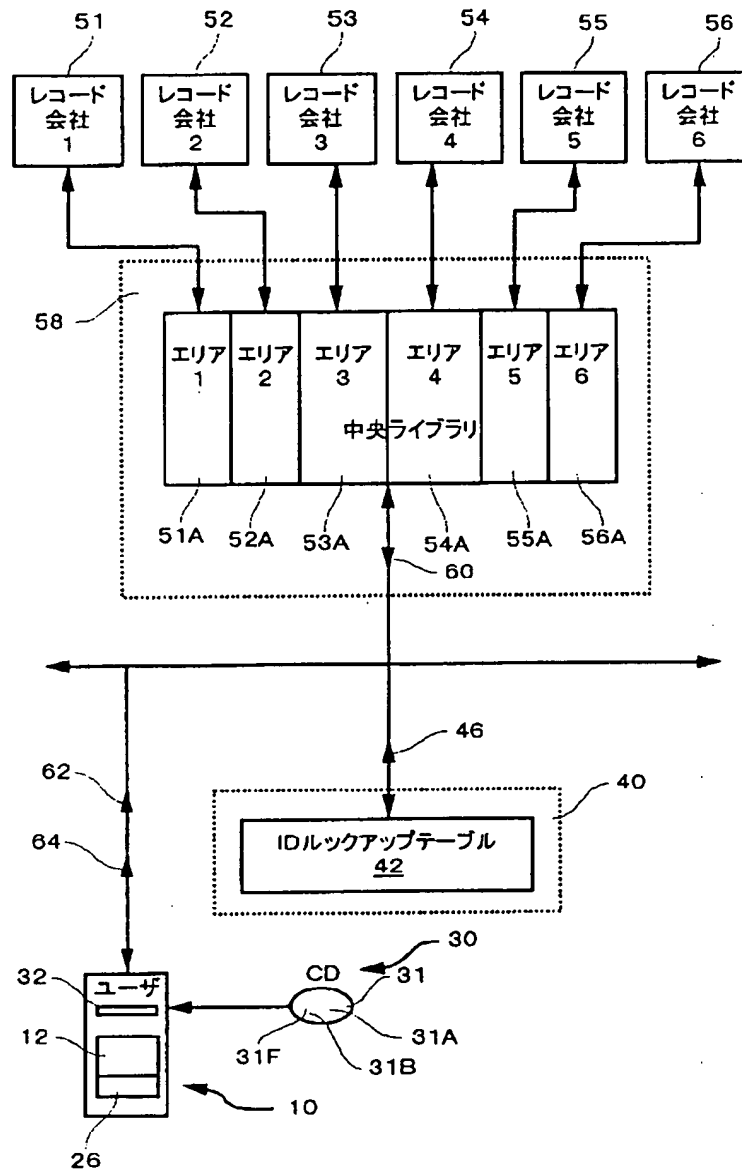
        matchtest = (cd2UID.id[0] == cdFromDBUID.id[0]) &&
(cd2UID.id[1] == cdFromDBUID.id[1]);

        printf ("The cd's match if result is non zero
matchresult=%d",matchtest);
        printf ("\n");
        printf ("\n");
    }

```

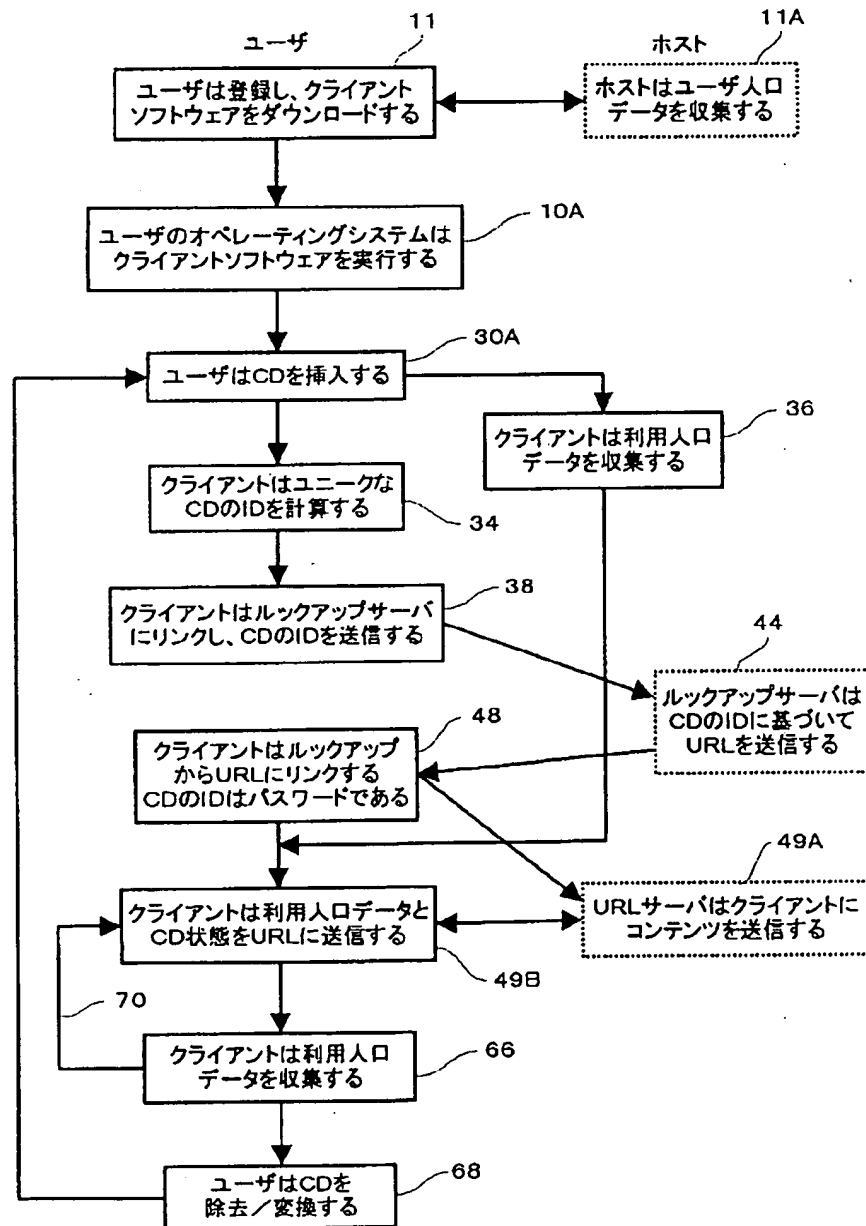
【図1】

図 1



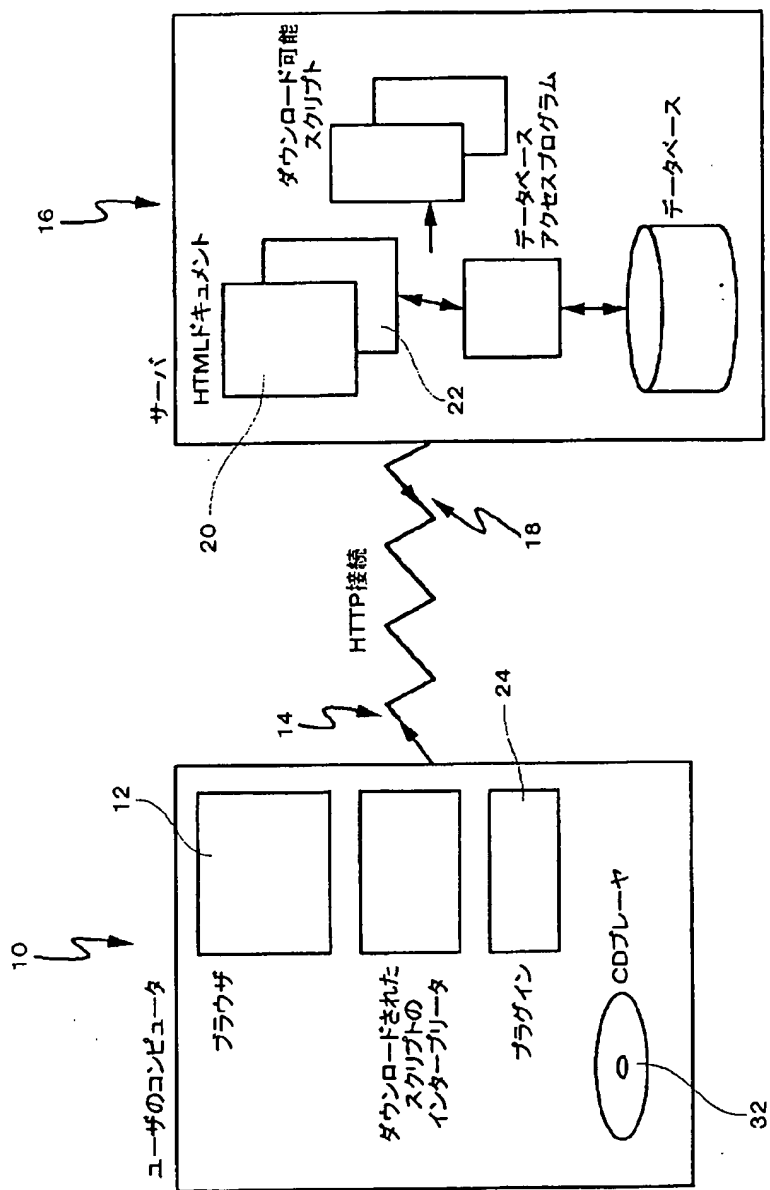
【図2】

図 2



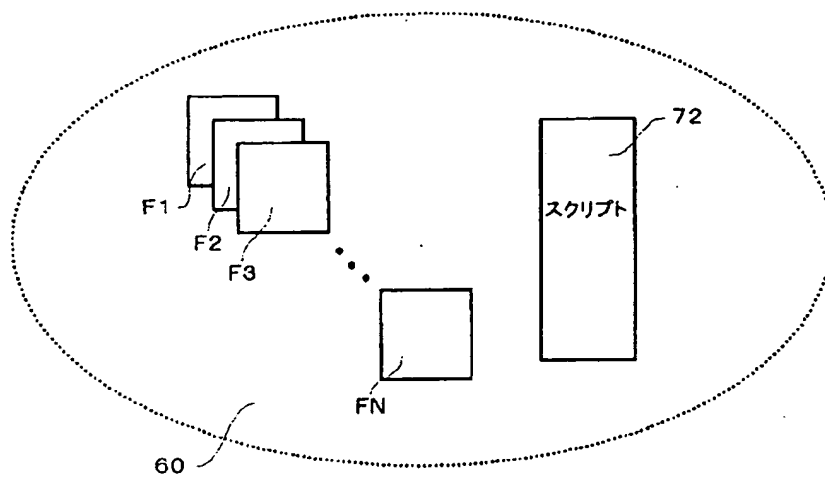
【図3】

図3



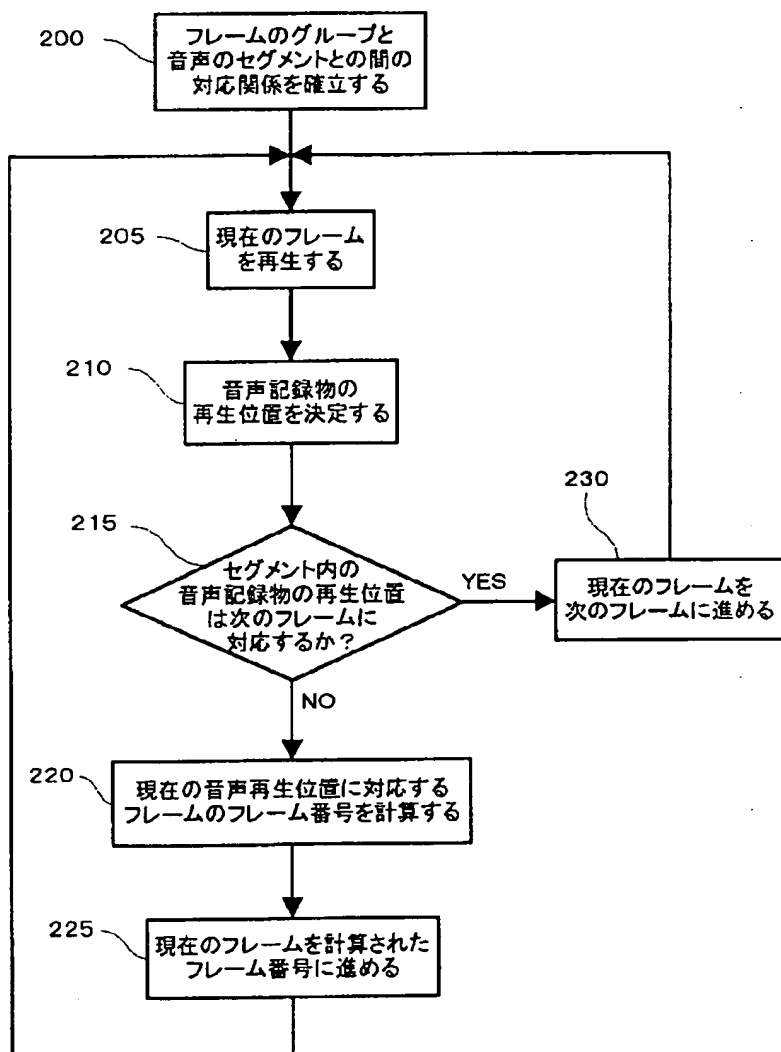
【図4】

図 4



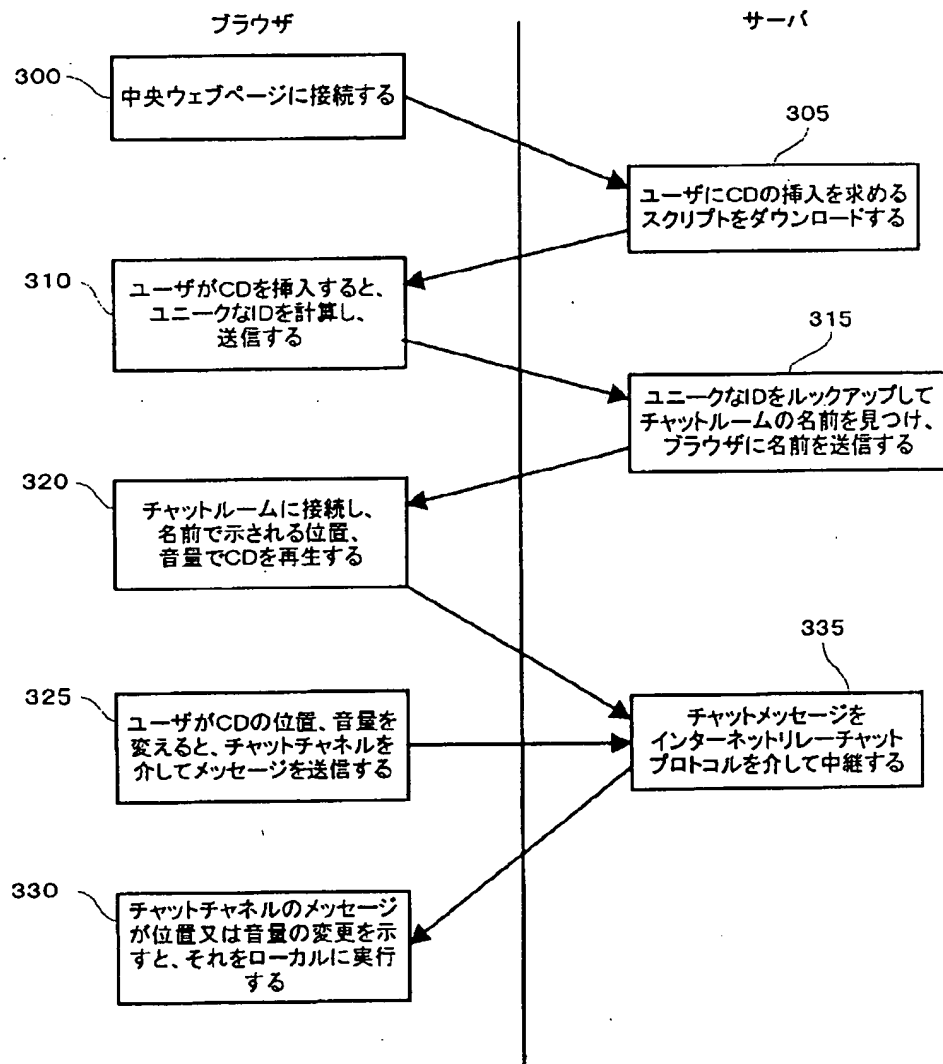
【図5】

図5



【図6】

図 6



INTERNATIONAL SEARCH REPORT

International Application No.
PCT/US 98/07660

A. CLASSIFICATION OF SUBJECT MATTER IPC 6 H04M11/08 H04L29/06 G06F17/30 G11B27/00 G07F17/30 H04H1/02		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) IPC 6 H04M H04L G06F G11B G07F H04H G10H		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practical, search terms used)		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y A	EP 0 194 143 A (SONY CORP) 10 September 1986 see abstract see page 2, line 11 - page 5, line 13 see page 8, line 14 - page 9, line 7 see page 10, line 16 - page 11, line 15 see page 12, line 28-30 ---	1,3-7,9, 11-17 2,10
Y A	WO 97 05616 A (SONY ELECTRONICS INC) 13 February 1997 see abstract see page 2, line 9 - line 17 see page 6, line 8 - page 7, line 29 see figure 2 --- -/--	1,3-7,9, 11-17 2,10
<input checked="" type="checkbox"/> Further documents are listed in the continuation of box C.		
<input checked="" type="checkbox"/> Patent family members are listed in annex.		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier document but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art. "G" document member of the same patent family		
Date of the actual completion of the international search		Date of mailing of the international search report
17 November 1998		26/11/1998
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel.: (+31-70) 340-2040, Tx. 31 031 epo nl, Fax: (+31-70) 340-3016		Authorized officer Llievens, K

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

International Application No.
PCT/US 98/07660

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>MASCHA M ET AL: "Interactive education: Transitioning CD-ROMs to the Web" COMPUTER NETWORKS AND ISDN SYSTEMS, vol. 27, no. 2, November 1994, page 267-272 XP004037997 see abstract see page 269, right-hand column, line 1 - page 271, right-hand column, line 16 -----</p>	<p>1,3-7,9, 11-17</p>

Form PCT/ISA/216 (continuation of second sheet) (July 1992)

フロントページの続き

(51) Int. Cl. ⁷	識別記号	F I	テーマコード (参考)
G 1 1 B 27/00		G 1 1 B 27/00	Z
27/10		27/10	Z
H 0 4 H 1/00		H 0 4 H 1/00	Z
H 0 4 N 7/173	6 1 0	H 0 4 N 7/173	6 1 0 A
// H 0 4 L 29/08		H 0 4 L 13/00	3 0 7 Z